

銘 傳 大 學

資 訊 工 程 學 系

專 題 研 究 總 審 文 件

本校一一三學年度 資訊工程學系

組員： 林錦輝、蔡文旭

吳睿騰、陳昱翔

所提專題研究： ChatGPT 在自主型代理人設計中的應用：

自主牌技學習

指 導 教 授： 王豐緒

中 華 民 國 一 一 三 年 十 二 月 九 日

# 中文題目全名

## 摘要

本研究探討 ChatGPT 在自主代理人設計中的應用，聚焦於自主牌技學習的實現。設計中結合多層**記憶模塊**，包括主要記憶流（規則）、環境記憶流（手牌資訊）、短期記憶流（每回合操作）、目標記憶流、計畫記憶流與反思記憶流模塊。透過代理人於回合結束後生成反思機制，強化其學習效果，逐步完善策略與能力。系統運行時，根據當前狀態，將相應的問題集及相關記憶模塊傳遞至 ChatGPT，實現高度智能化與自主化。

**關鍵字：**自主代理人、記憶模塊、反思

# 目錄

摘要 .....	I
目錄 .....	II
圖目錄 .....	IV
第一章 緒論 .....	1
1.1 研究動機 .....	1
1.2 研究目的 .....	1
第二章 文獻探討 .....	2
2.1 記憶流 .....	2
2.2 評估方法[2] .....	3
2.3 如何下指令（prompt）[3] .....	3
第三章 研究方法 .....	5
3.1 ChatGPT API .....	5
3.1.1 Request .....	5
3.1.2 Response .....	7
3.2 代理人設計 .....	7
3.2.1 自主型代理人設計 .....	8

3.2.2	策略導向代理人設計 .....	8
3.3	系統設計 .....	9
第四章	系統實作與測試 .....	10
4.1	系統介紹 .....	10
4.2	麻將設計 .....	10
4.3	實驗設計 - 目標導向 .....	13
4.4	實驗結果 .....	14
第五章	結論 .....	17
參考文獻	.....	18

## 圖目錄

圖 3-1 Request API 範例 .....	5
圖 3-2 Messages 範例 .....	6
圖 3-3 ChatGPT 對應 Messages 內容範例 .....	6
圖 3-4 Request API 說明 .....	6
圖 3-5 Response 範例 .....	7
圖 3-6 取值範例 .....	7
圖 3-7 自主型代理人架構 .....	8
圖 3-8 系統架構圖 .....	9
圖 4- 1 麻將程式類別圖 .....	12
圖 4- 2 麻將程式流程圖 .....	13
圖 4- 3 最佳出牌 .....	14
圖 4- 4 幫下家贏牌 .....	14
圖 4-5 GPT 胡言亂語 .....	15
圖 4-6 10 局的勝負 .....	15
圖 4-7 GPT-4o 丟牌分布 .....	16
圖 4-8 GPT-4o-mini 丟牌分布 .....	16
圖 4-9 GPT-3.5-turbo 丟牌分布 .....	16

# 第一章 緒論

## 1.1 研究動機

在遊戲開發中，有多種方法用於驅動遊戲 AI，例如行為樹、有限狀態機等。而 LLM（大型語言模型）的應用則涵蓋對話機器人、AI Vtuber 等場景，但以 LLM 驅動遊戲 AI 的案例仍然相對罕見。

在《Generative Agents》[1]中，基於 GPT 的代理人能夠透過模擬人類行為，生成接近人類的互動模式。人類社會中，學習是一項核心能力，人類可以通過反思經驗，調整行為模式，從而形成新技能或洞見。如果能將學習機制引入到代理人中，這些代理人不僅能模仿行為，還能自主適應動態環境，甚至發展出全新的行為策略。

因此，我們的研究動機是探索如何賦予 GPT 驅動的代理人學習能力，從而進一步提升其模擬人類行為的適應性與創造性。

## 1.2 研究目的

本研究旨在探討 ChatGPT 技術如何在麻將遊戲中模擬人類的打牌行為，以及其作為智能代理的表現。具體，我們將尋求以下目標：

- 1.ChatGPT 是否能根據遊戲規則、場上牌局資訊以及歷史記錄，作出符合人類思維模式的決策
- 2.ChatGPT 能否透過多次遊戲或模擬，逐漸改善其打牌策略

3.ChatGPT 的行為是否能被人類玩家接受以勝率、決策效率多樣性等指標評估 ChatGPT 作為麻將智能代理的整體表現。

## 第二章 文獻探討

### 2.1 記憶流

生成式代理人 (Generative Agents) 中的記憶流 (Memory Stream) 是模擬人類長期記憶的核心技術，用於儲存代理人過往經驗的完整記錄，並支持其行為生成。記憶流由一系列記憶物件組成，每個記憶物件包含自然語言描述、建立時間戳記以及最近存取時間戳記。

記憶流的基本單元是**觀察** (Observations)，即代理人感知到的事件，包括自身行為或與其他角色的互動。這種結構的主要目的是幫助代理人在不同情境下進行合理推理，從而模擬人類的決策過程。然而，由於生成模型的上下文窗口有限，完整的記憶流可能過於龐大且難以處理，因此需要設計高效的記憶提取與篩選機制，以保留與當前場景相關的內容，提升運算效率。

這種設計不僅增強了代理人對長期互動的一致性維持能力，還能模擬多場景間的行為連貫性。記憶流已被廣泛應用於自然語言處理、遊戲設計及多代理人互動系統中，特別是在模擬人類行為方面展現了極大的潛力。

未來的研究可專注於開發更高效的記憶提取算法與動態更新策略，結合生成式 AI 和強化學習技術，打造精確且具高度適應性的代理人系統。

## 2.2 評估方法[2]

由於臺灣麻將本身具有高度的隨機性，客觀評估人工智慧代理的性能相對困難。為了更準確地評估 ChatGPT 以及所提出方法的性能，我們採用了獲勝次數作為主要性能指標，而非比賽中常用的得分。這是因為在自摸情況下，得分容易受到運氣的影響，而獲勝次數則能更有效地反映代理的實力。

在比較兩種不同方法或人工智慧代理時，每個代理均由兩個實例構成遊戲的四名玩家。例如，在比較方法 A 和方法 B 時，代理 H1 使用方法 A，而 H2、H3 和 H4 則使用方法 B。四個代理按照 H1 - H2 - H3 - H4 的順序入座。每次實驗中，並仔細記錄每個代理的勝場數和放槍次數。

## 2.3 如何下指令（prompt）[3]

與 AI 有效互動的關鍵在於**清晰明確**的指令，這能大幅提高輸出的品質。Jeff Su 在文章中提出了**六大要素**，幫助使用者與 AI 高效溝通：

1. **任務**：作為指令的核心，應以動詞開頭，明確表達需求。例如：

「請給我一個\_\_\_\_\_」或「請設計一個\_\_\_\_\_」。同時，保持禮

貌有助於促進良好的互動。

2. **脈絡**：提供必要的背景資訊，包括任務目標、理想結果和環境

限制，但避免過多細節，避免讓 AI 分心。

3. **範例**：為 AI 提供具體的參考模板，讓生成內容更接近使用者

的預期。

4. **指定角色**：引導 AI 模擬特定角色的視角，例如「行銷專家」或「文案編輯」，以解決問題或提供建議。
5. **輸出格式**：明確指定內容的結構，如段落、表格或標題格式，確保結果符合需求。
6. **語調**：賦予指令特定的情感色彩，透過關鍵詞或描述，讓生成的文字更具感染力和共鳴。

這六個要素的重要性依次為：**任務 > 脈絡、範例 > 角色、格式、語調**。Jeff Su 提出的公式為：

高品質指令 = [角色] + [脈絡] + [任務] + [格式] + [範例] + [語調]

此公式幫助使用者組織資訊，使 AI 更精確理解需求，進而產出高品質的結果。

## 第三章 研究方法

### 3.1 ChatGPT API

#### 3.1.1 Request

圖 3-1 展示了一個使用 cURL 發送請求至 OpenAI API 的範例。此範例通過 /v1/chat/completions 路徑向聊天模型請求回應，具體操作簡潔明了。

圖 3-2 則詳細展示了 Messages 的內容，說明了如何組織和傳遞消息以供模型生成適當的回應。

最後，圖 3-3 進一步解釋了每個屬性所代表的含義，幫助使用者更全面地了解 API 請求與回應的結構和功能。

```
curl https://api.openai.com/v1/chat/completions \
-H "Content-Type: application/json" \
-H "Authorization: Bearer $OPENAI_API_KEY" \
-d '{
  "model": "gpt-4o-mini",
  "messages": [],
  "response_format": {
    "type": "text"
  },
  "temperature": 1,
  "max_tokens": 256,
  "top_p": 1,
  "frequency_penalty": 0,
  "presence_penalty": 0
}'
```

圖 3-1 Request API 範例

```
"messages": [
  {
    "role": "user",
    "content": [
      {
        "type": "text",
        "text": "hello"
      }
    ]
  },
  {
    "role": "assistant",
    "content": [
      {
        "type": "text",
        "text": "Hi there! How can I assist you today? 😊"
      }
    ]
  }
]
```

圖 3-2 Messages 範例

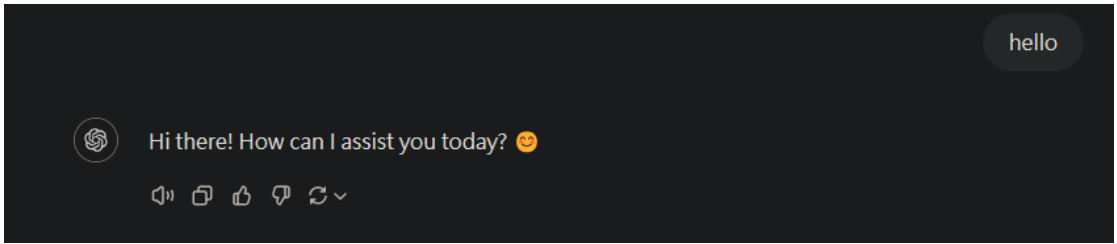


圖 3-3 ChatGPT 對應 Messages 內容範例

部分	欄位/屬性	值	說明
Request Header	POST	/v1/chat/completions	請求的 API 路徑，表示執行聊天模型生成操作。
	Host	api.openai.com	目標伺服器的主機名。
	Content-Type	application/json	請求內容的格式為 JSON。
	Authorization	Bearer \$OPENAI_API_KEY	用於身份驗證的 API 金鑰，\$OPENAI_API_KEY 需替換為實際密鑰。
Request Body	model		指定使用的模型。
	messages		可加入對話內容如用戶問題。
	response_format		指定回應的格式。
	temperature	0~1	控制生成文本的隨機性，值越高生成結果越多樣化。
	max_tokens		限制生成回應的最大 token 數（文本長度）。
	top_p	0~1	使用核採樣方法的設定，值為 1 表示不啟用核採樣。
	frequency_penalty	0~2	控制回應中重複內容的懲罰，值越高越少重複。
	presence_penalty	0~2	鼓勵生成新內容的懲罰，值越高越多新話題。

圖 3-4 Request API 說明

### 3.1.2 Response

根據圖 3-5 的 ChatGPT API 回應範例，我們可以看到生成的文字儲存在 `choices[0]["message"]["content"]` 中。圖 3-6 展示了

提取該文字的程式碼範例，可以成功擷取回傳內容。

```
{
  "id": "chatcmpl-Ab9s13zkTqcsZLTGPTt4Hwj2lZEEdt",
  "object": "chat.completion",
  "created": 1733419441,
  "model": "gpt-4o-2024-08-06",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "Hello! How can I assist you today?",
        "refusal": <null>
      },
      "logprobs": <null>,
      "finish_reason": "stop"
    }
  ],
  "usage": {
```

圖 3-5 Response 範例

```
val = response["choices"][0]["message"]["content"]
```

圖 3-6 取值範例

## 3.2 代理人設計

本節中將說明代理人設計，並拆成 2 部分討論，分別是自主型代理人設計、策略導向代理人設計。

### 3.2.1 自主型代理人設計

為了實現自主型代理人的設計目標，我們採用了如圖 3-7 所示的代理人架構[]，模擬人類的思考模式。該架構透過感知模組來觀測環境狀態，並檢索記憶流以支持計畫的制定和反思的生成。代理人執行行動後，透過持續加入每次反思的結果，逐步實現自主學習的能力。

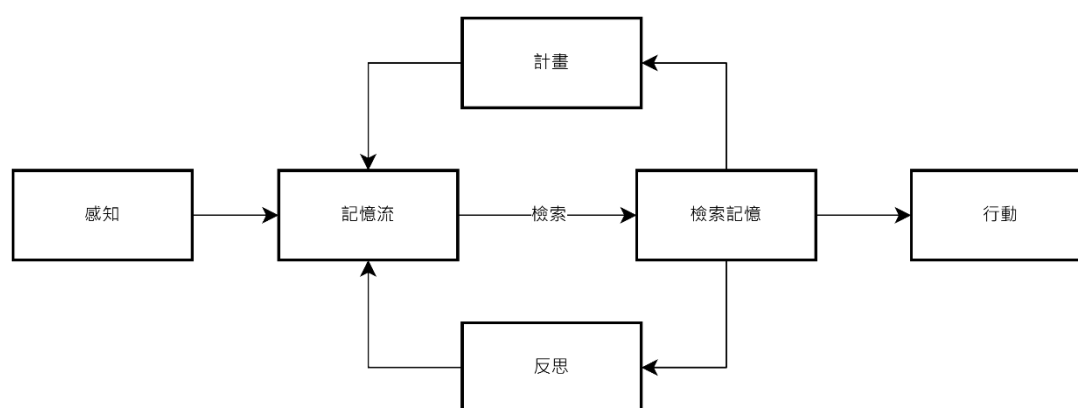


圖 3-7 自主型代理人架構

### 3.2.2 策略導向代理人設計

為了評估自主型代理人的性能，我們設計了一個策略導向代理人作為對手。該代理人採用了一套簡單的策略，包括「丟沒組合」、「丟字」以及「丟邊」，作為其核心行為模式。

### 3.3 系統設計

本系統以 Godot 4.2 作為執行環境，並採用 GDScript 程式語言設計了一款麻將遊戲及其代理人架構。在系統中，遊戲相關數據以數據形式傳遞給代理人進行決策處理。策略導向代理人通過計算接收到的數據，生成並執行預定策略。自主型代理人則透過 OpenAI 提供的 ChatGPT API，對數據進行分析，生成計畫與反思，並依據計畫執行相應的行動。系統架構如圖 3-8 所示。

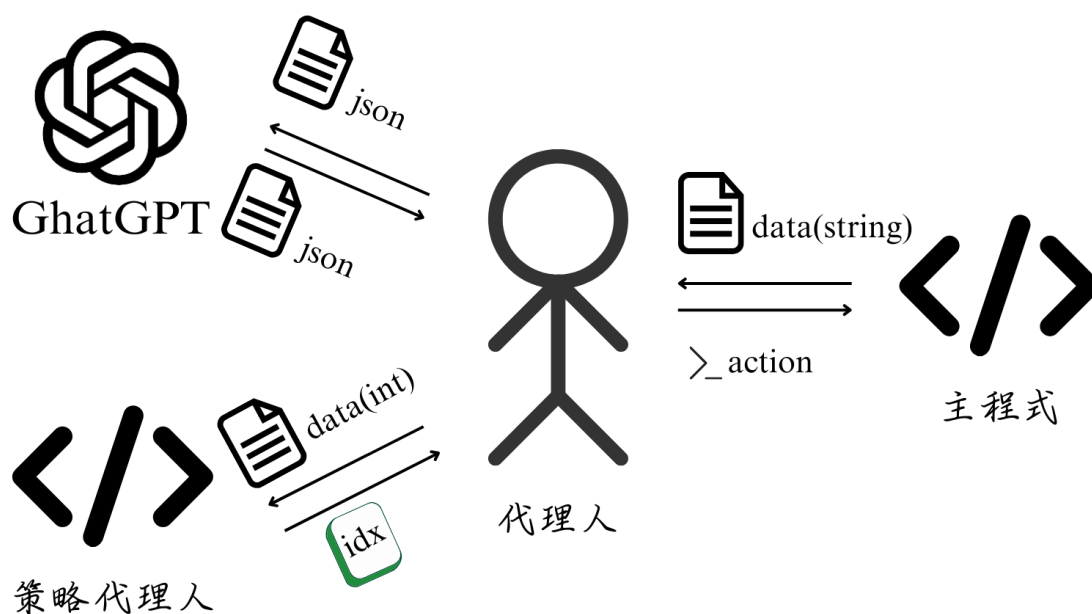


圖 3-8 系統架構圖

## 第四章 系統實作與測試

### 4.1 系統介紹

為了測試由 ChatGPT 驅動的自主代理人在環境改變時是否能做出不同決策，我們選擇了麻將作為實驗基礎。麻將具有較高的隨機性，且需要一定的推理能力，能有效檢驗代理人的適應性與策略性。

### 4.2 麻將設計

我們在設計麻將時，為了能夠清晰的表達遊戲製作的優先順序，為其我們設計的 2 張圖輔助編輯程式。

如圖 4-1 所示，我們先為期擬定了張類別圖，好為其去設計幾個重要的類別，我們將其側重於資料的交換，模擬著不同代理人所應該擁有的功能以及互相的交互。

圖 4-2 表達著程式的整體流向，也就是模擬麻將運行所擬定下來的敘述。麻將遊戲的設計需要結合遞歸規則與特定中止條件，確保在遊戲中能正確處理牌組的分解與檢查，並完成合理的判斷。以下是此麻將設計的核心規則與運作細節：

#### 中止條件

遞歸處理的結束基於以下兩個條件：

- 1.無效構牌數大於 2：當剩下的牌無法構成有效的牌型時，遞歸即停止。
- 2.檢查的牌數大於等於手牌長度：當已檢查過的牌數達到

或超過整副手牌的數量時，表示已無剩餘牌可進一步操作，遞歸停止。

### 遞歸條件

為實現麻將的牌型組合判斷，遞歸過程中需要滿足以下條件之一：

僅構成一對「眼睛」：即牌型中至少需要一對相同的牌作為基礎。

構成「碰」：三張相同的牌，例如三個「東風」。

構成「吃」：三張連續的數字牌，例如「3 萬」、「4 萬」、「5 萬」。

不參與組合的牌數 +1：處理過程中若有牌未能構成任何組合，則遞歸進一步嘗試其他組合方式，直至符合條件。

流程圖說明，

在設計中，遊戲的運作流程可以用以下步驟概述：

遊戲初始化

將牌局初始化，準備所有玩家的手牌與牌堆。

迴圈控制

當遊戲未結束時（`end == false`），進行玩家操作，包括抽牌、打牌等。

玩家回合操作

玩家選擇是否進行抽牌操作。

若選擇抽牌，從牌堆中隨機獲取一張牌。

檢查抽牌後是否滿足遊戲的贏牌條件。

若不抽牌，進行打牌操作，並將打出的牌放入公共區域。

## 牌型檢查與處理

系統檢查玩家打出的牌是否能構成其他玩家的「吃」、「碰」或「槓」。

若能觸發相關規則，則由觸發玩家完成相應操作。

## 遊戲結束條件

當任一玩家滿足贏牌條件時（例如胡牌），或當牌堆無剩餘牌時，遊戲結束。

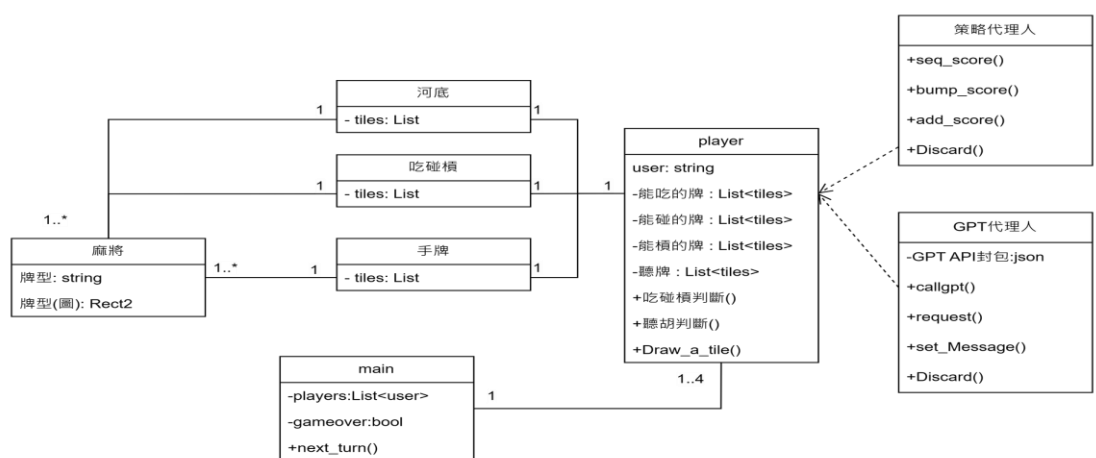


圖 4- 1 麻將程式類別圖

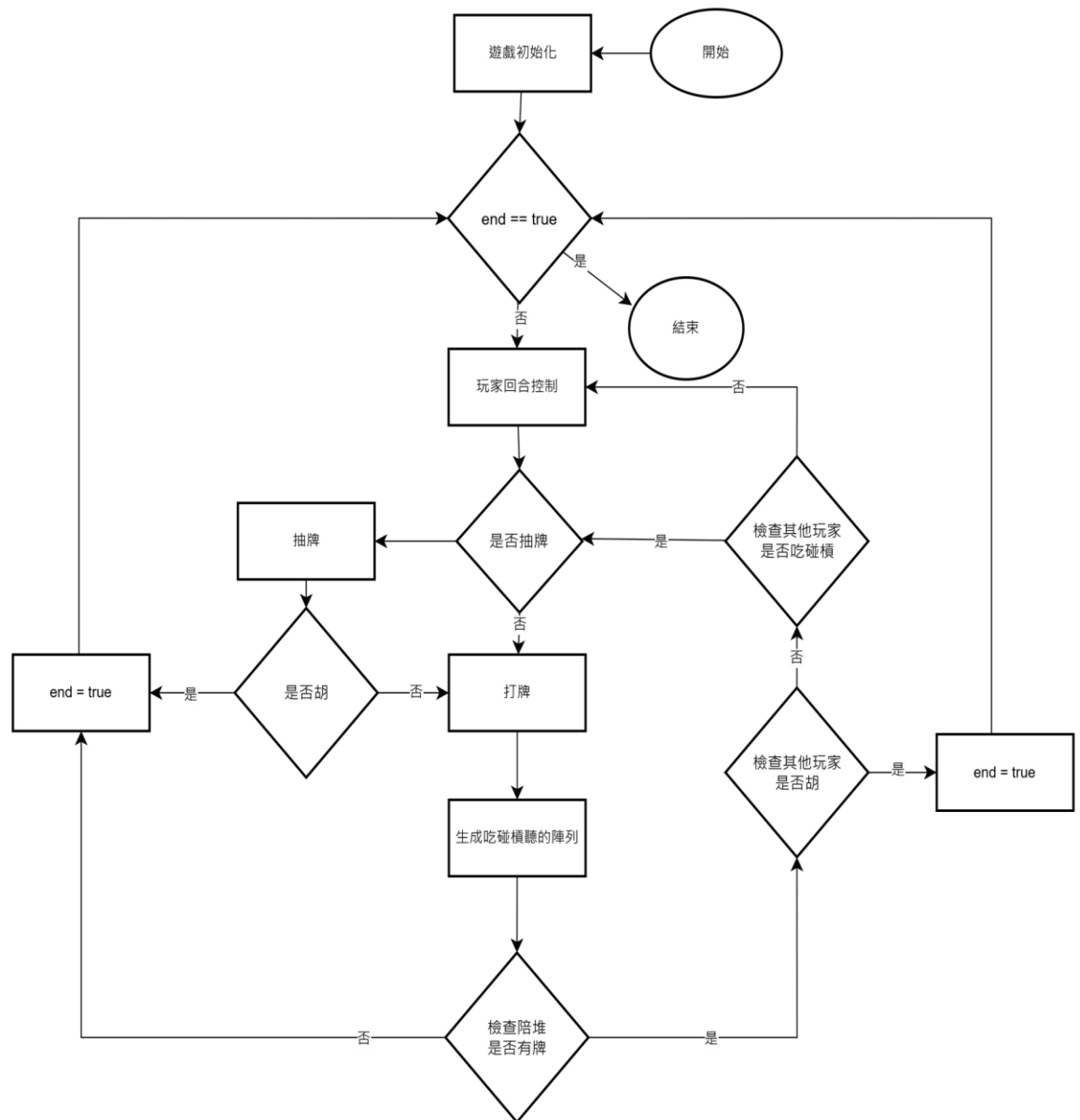


圖 4- 2 麻將程式流程圖

### 4.3 實驗設計 – 目標導向

既然是目標導向，那我們更換目標的內容是否能夠產生出新的策略此實驗我們觀察其計畫是否符合目標，圖 4-3 最佳出牌、圖 4-4 幫下家贏牌，如下圖所示，GPT 能夠靈活的應對不同的目標給出不同的策略。

```

根據你的手牌和當前牌局，我們需要制定一個最佳的出牌策略來提高胡牌的機會。以下是我的計劃和思考過程：

### 手牌分析
- **現有手牌**：四萬，五萬，六萬，六筒，六筒，七條，八條，二條
- **已經形成的組合**：
  - 四萬，五萬，六萬：這是一個順子
  - 六筒，六筒：這是一個對子，目前可以作為眼睛或潛在的刻子

### 潛在組合
- **條子部分**：
  - 七條，八條：這是接近順子的組合，可以等六條或九條來完成順子
  - 二條：與其他條子不易組成順子，考慮丟棄

- **筒子部分**：
  - 六筒，六筒：目前為一對，但如果有機會，可以碰成刻子

### 對手行動考量
- **上家**：已經吃碰槓了多組筒子，可能需要筒子
- **對家**：沒有明確的筒子組合，筒子可能不需要
- **下家**：已經槓了一組六萬，條子方面可能不易組成組合

### 出牌策略
考慮到以上因素，二條不太可能與其他牌組成有用的順子或刻子，而其他牌在各自的花色中更有成為順子或刻子的潛力。因此，最佳的選擇是丟棄二條。

### 計畫總結
1. **丟棄牌**：二條
2. **保留和觀察**：專注於筒子的可能性，尤其是碰成刻子的機會，以及條子部分等待六條或九條來完成順子。

```

圖 4- 3 最佳出牌

```

為了讓下家贏牌，我需要考慮如何將對下家最有利的牌留在場上，並丟棄對他們構成威脅或無用的牌。以下是我的計畫：

1. **觀察下家的可能需求**：下家已經吃碰槓了七條、八條、九條，這意味著他們可能在尋找條子的其他順子或刻子。
2. **注意其他玩家的動作**：上家和對家的出牌可幫助我判斷剩餘牌的情況。上家和對家都未出過條子，這意味著在下家手中可能需要條子。
3. **選擇丟棄的牌**：考慮到下家已經打出過南、東、北等風牌，那些牌對他們可能無用，因此我應該避免丟出條子，特別是二條、三條、四條這些可能對下家有用的牌。

綜合以上分析，我選擇丟棄一張手牌中對局勢影響較小的風牌。這樣可以避免干擾下家的計畫，並且沒有助長其他玩家的機會。基於這個計畫，我選擇丟棄「北」。

```

圖 4- 4 幫下家贏牌

## 4.4 實驗結果

在系統運行過程中，我們可以觀察到其為了打牌所生成的計畫。然而，計畫中經常出現語意不清或不合理的情況，這種現象在三個模型中皆有發生。尤其是 GPT-4o-mini 模型，經常試圖打出其他玩家已打出的牌（如圖 4 -5 所示）。相比之下，其他兩個模型則較少出現這類問題，但仍偶爾會作出奇怪的決策。儘管如此，系統在環境適應能力方面表現優異，能夠根據場上的變化採取不同的策略，展現出一定的靈活性與應變能力。

你的手牌是：

- 一萬，二萬，三萬，六萬，三筒，七筒，九筒，五條，八條，東，南，西，西，北，發，中，四筒

上家的打牌是白，下家的打牌是南。你的手牌中有一些已經組成的順子或刻子潛力，也有一些不太可能組成有利組合的牌。

考慮到以下幾個因素：

1. **\*\*順子的潛力\*\***：一萬、二萬、三萬可以組成順子，六萬可以與五條、七筒組成其他潛在的順子。
2. **\*\*刻子的潛力\*\***：手中有南、東、西、北、發、中等風牌，但目前沒有成組的情況。
3. **\*\*無法組成的牌\*\***：白是無法組成任何有利的組合的牌，因此不需要保留。

基於以上考量，建議丟棄的牌是 **\*\*白\*\***。這不會影響你的組牌潛力，並且可以讓你保留其他更有用的牌。

圖 4- 5 GPT 胡言亂語

如圖 4-6 所示，由於麻將中隨機因素占比較高，僅以 10 局對局結果難以有效比較三個模型的麻將技巧與性能。因此，我們採用記錄丟牌分布的方式，分析三個模型在丟牌時的策略性，從而判斷其行為是否具備規律性或僅為隨機選擇。

在圖 4-7、圖 4-8 及圖 4-9 中可以觀察到，三個模型在丟牌策略上均呈現出優先打出字牌與邊牌的傾向，表明其行為具有一定的策略性，而非完全隨機。

模型	勝場(贏牌)	敗場(放槍)	局數
ChatGPT-4o	0	2	10
ChatGPT-4o-mini	0	3	10
ChatGPT-3.5-turbo	1	4	10

圖 4-6 10 局的勝負

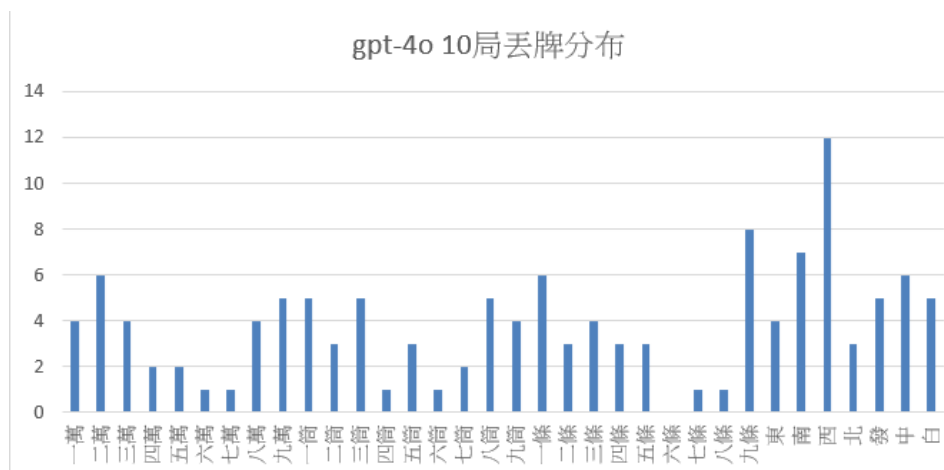


圖 4-7 GPT-4o 丟牌分布

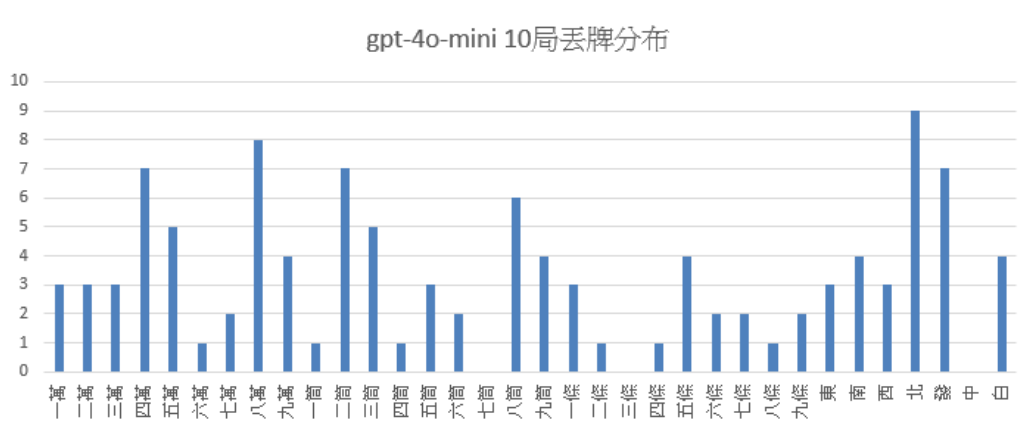


圖 4-8 GPT-4o-mini 丟牌分布

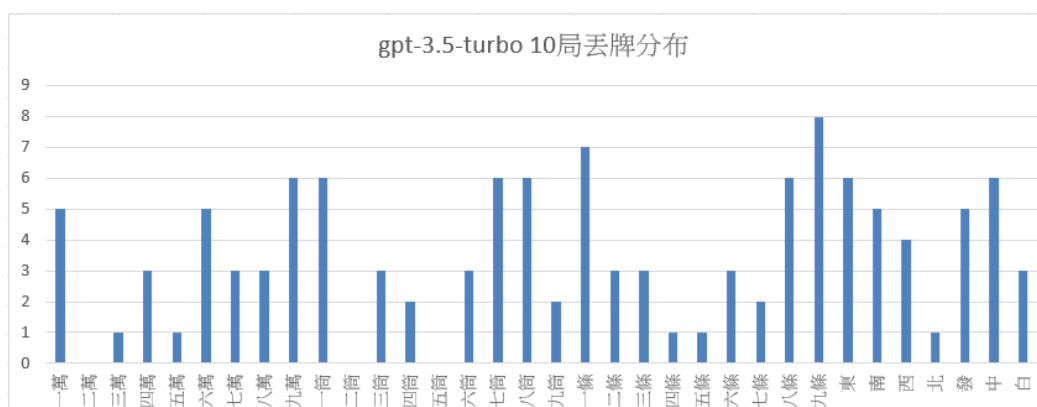


圖 4-9 GPT-3.5-turbo 丟牌分布

## 第五章結論

本專題旨在探討由 ChatGPT 驅動的代理人是否具備一定的適應力和學習力。研究結果顯示，適應力的表現與預期相符。由於 LLM（大型語言模型）本身依賴使用者的輸入來生成新的回應，其適應能力表現出色，能夠根據需求快速調整回應方式。

然而，對於學習力的觀察則受到遊戲中運氣成分較大的限制，難以明確評估代理人在經過反思後是否能通過改進策略獲得更高的勝率。因此，目前尚無法確定其學習能力是否能在在此情境下展現出顯著的效果。

當前階段，我們認為應該採用複合式 AI 的方式，讓 LLM 專注於語言處理，而將遊戲 AI 的部分交給深度學習技術，兩者協同合作以構建更優秀的遊戲 AI。隨著 LLM 技術的持續突破和成本的下降，我們認為這樣的項目未來將更具價值。

## 參考文獻

[1]. Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, Michael S. Bernstein, Generative Agents: Interactive Simulacra of Human Behavior

[2] 麻將 AI 對局 - Artificial-Intelligence - 資訊工程 | Hsing Yu == We are the dreamers. We make it happen 'cause we can see it.

<https://hsingyu.blog/computer-science/Artificial-Intelligence/%E9%BA%BB%E5%B0%87%20AI%20%E5%B0%8D%E5%B1%80/>

[3]<https://vocus.cc/article/663dc557fd89780001d9dfdd>